

PSKmail server

1.0.34

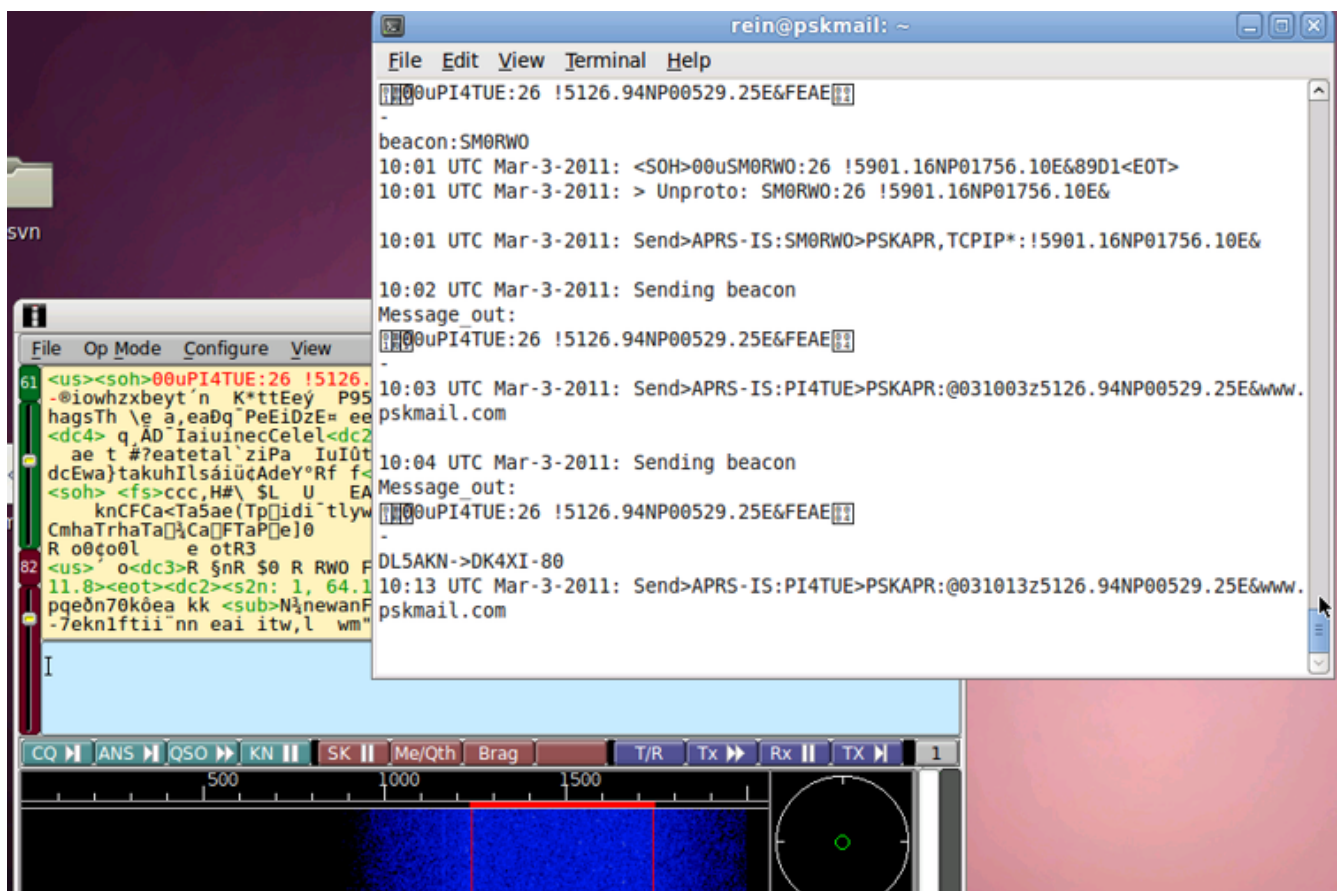


Table of Contents

PSKmail server	1
What is PSKmail?.....	3
So, what makes PSKmail special?.....	3
In summary, what is pskmail?.....	4
The PSKmail server.....	4
Hardware selection.....	5
Transceiver.....	5
Soundcard interface.....	5
Server installation.....	6
Basic installation procedure.....	6
Install fldigi for Linux on UBUNTU.....	6
Configure fldigi for use with the PSKmail server.....	7
Carrier frequency (Hz).....	7
AFC range (Hz).....	7
Reset to Carrier and Report ARQ frames average S/N.....	7
RS ID.....	8
Installing the PSKmail server on UBUNTU.....	9
Getting Puppy Linux.....	10
Installing fldigi on Puppy Linux.....	10
Installing the server on puppy.....	10
Server configuration.....	11
The pskmailrc.pl file.....	11
The freq.txt file.....	12
Running the server.....	13
Running the server remotely.....	13
Time keeping.....	13
Implemented services.....	14
Sending bulletins.....	14
Sending long bulletins.....	14
Sending weather beacons.....	14
Server commands.....	15
Appendix A, Mode selection.....	17
Mode tables for adaptive asymmetric link control.....	17
APRS Mode choice.....	18
Appendix B:.....	19
PSKmail ARQ protocol specification (2nd draft).....	19
Unproto (non-ARQ) operation.....	23
References.....	24

What is PSKmail?

PSKmail is a system that uses data modes to enable clients to access both the internet and to communicate with other pskmail clients. Data modes are used over a suitable radio, mostly its an hf radio but higher frequencies are also in use.

Just saying internet access could mean that the system should support a whole range of services, and indeed it does just that:

1. Messaging
 - Handles the users regular email account, including gmail etc.
 - For emergencies its even possible to handle messaging off the internet (the whole internet could be down, messaging works anyway)
2. Web browsing, fetch and read any web page
3. Twitter, send and read tweets
4. File up and download
5. Supports APRS including position reports, messages and short emails.
6. Supports map clients like Xastir and Uiview

But, that's just part of what's there now. Servers can also fetch your weather report periodically, show you where the nearest camp site is and tell what other users are nearby and so on...

So, what makes PSKmail special?

PSKmail is designed with the mobile/portable (maritime too) ham in mind. There is no big, power hungry and expensive modem to carry along. PSKmail is designed to use your pc soundcard as modem, right now the digital modem program Fldigi is what is being used for that very purpose.

A mobile/portable station is mostly not what we sometimes refer to as a “big gun”, antennas and power output tend to be on the modest side. With that in mind PSKmail uses bandwidth efficient modes, hardly ever wider than 500 Hz (for higher frequencies that may of course be the case).

Wide data modes require a better signal to noise ratio than narrow ones, PSKmail is even able to adapt the speed and the mode so that the client can select a really narrow and efficient mode. Then again as the signal to noise ratio increases or the qrm goes away it will use wider and faster modes.

PSKmail network topology is flexible:

1. Clients can connect each other using ARQ, for chat of message transfer
2. Clients can connect to servers, which are gateways to the internet
3. Clients can send each other messages in unconnected mode
4. Clients can send APRS message and position beacons to the APRS backbone through the servers
5. Clients also contain a passive Igate, which can gateway messages and posits to APRS
6. Clients contain a telnet port where you can connect a map application like Xastir or UIview

In summary, what is pskmail?

The main characteristics of PSKmail can be summed up as:

- A semi duplex ARQ Chat client capable of transferring files
- A full fledged solution for HF APRS, including integration with the map clients
- An internet connection that can be used where there is no internet connection. Such as from a boat at sea or a cottage somewhere or on expedition or...
- Narrow, efficient, data modes are used. Thus enabling even the most modest setup.
- Servers on short wave provide long range connectivity
- No expensive and bulky modem, all you need is a PC, a transceiver and an antenna
- Available on multiple operating systems (Linux and Windows, MacOS and FreeBSD)

These features make it especially valuable for emergency communications.

The PSKmail server

The Pskmail server functions as a gateway between the PSKmail network and the Internet. The most important internet services it offers are:

1. Email send and receive
2. Browsing the web (text only)
3. APRS services (messaging, short emails, posits)
4. Twitter

The server can interface with a local Email server in case the internet is lost in an emergency. It holds an info file store for download, which can be periodically updated

The PSKmail server can also be used to broadcast bulletins, when necessary.

The server software is only available for Linux, but there are provisions to easily run it on a windows computer...

Hardware selection

Transceiver

There are many kinds of transceivers in use, many work really well with pskmail. Selecting a transceiver can be tricky and is of course guided by many factors. A few words of advice may be helpful for the beginner anyway.

A good transceiver should have an optimized digital mode, a mode where a narrow filter can be used for better reception. Adding a 500 Hz narrow filter is a very good way to enhance the S/N ratio on receive and may be the difference between a working and a non working connection.

Older transceivers may lack a dedicated digital mode and may require special handling in order to employ a narrow filter. One way around that is to use a split and listen in cw, with filter, and transmit in usb. That method may also require tweaking the rig BFO for cw. A memory for all of that is then useful.

A server that intends to use some kind of scanning will require a computer control connector, such as CI-V or CAT or whatever hamlib will support. Computer control is not necessary for a client at the moment but could be useful for handling the PTT.

Soundcard interface

A soundcard interface is used to connect the computer to the transceiver. It may in theory be possible to connect straight lines between sound in/out and the microphone connector on the transceiver but that method has so many drawbacks that it is not recommended. A soundcard interface will provide isolation between the pc and the radio, it will normally be attached to the transceiver data connector and that helps keep the receive level correct. Its also much better to have the interface control the keying of the PTT instead of using a VOX, although the timing constraints of pskmail will also support VOX if that is easier.

Server installation

First you have to take a decision with regard to the operating system to be used. The PSKmail server was designed to run on Linux, and there are no plans to port it to windows.

There are basically 3 flavours:

If you are already familiar with Linux, you probably want to use your favourite Linux distribution, we have tested on Debian, Gentoo, and of course UBUNTU.

If you are not familiar with Linux, and you only have a windows machine, you should opt for a solution with Puppy Linux. Puppy can run from a Live CD or from a USB stick. This is a perfect solution for deploying an ad hoc server, as you can use any old hardware to run it. Moreover, the puppy version comes basically pre-configured...

You can also run the server in a virtual machine like VMWare, Virtualbox or QEMU.

Basic installation procedure

I get a lot of emails saying “I installed everything and hit the connect button, but nothing happens. PSKmail does not work”.

To prevent this, please follow the recommended install procedure...

1. Install fldigi for Linux
2. Configure fldigi for use with the PSKmail server
3. Check the PTT and audio connections to your transceiver
4. Test fldigi with your transceiver with PSK500 and THOR8
5. Install the PSKmail server
6. If you want a scanning server, test if hamlib works properly with your transceiver
7. Configure the server for your station (call, smtp server, frequency file)
8. Test the server with an external client

Install fldigi for Linux on UBUNTU

Download fldigi from: <http://hermes.esrac.ele.tue.nl/pskmail/fldigi/>

or from <http://www.w1hkj.com/download.html>

Follow the installation instructions on <http://www.w1hkj.com/FldigiHelp-3.21/New-Install.html>

Configure fldigi for use with the PSKmail server

PSKMail needs a modem to work, that modem is Fldigi and setting it up properly is absolutely crucial. General Fldigi setup is best described by the help provided by the Fldigi project. But, there are a few things that need to be taken care of for pskmail.

Carrier frequency (Hz)

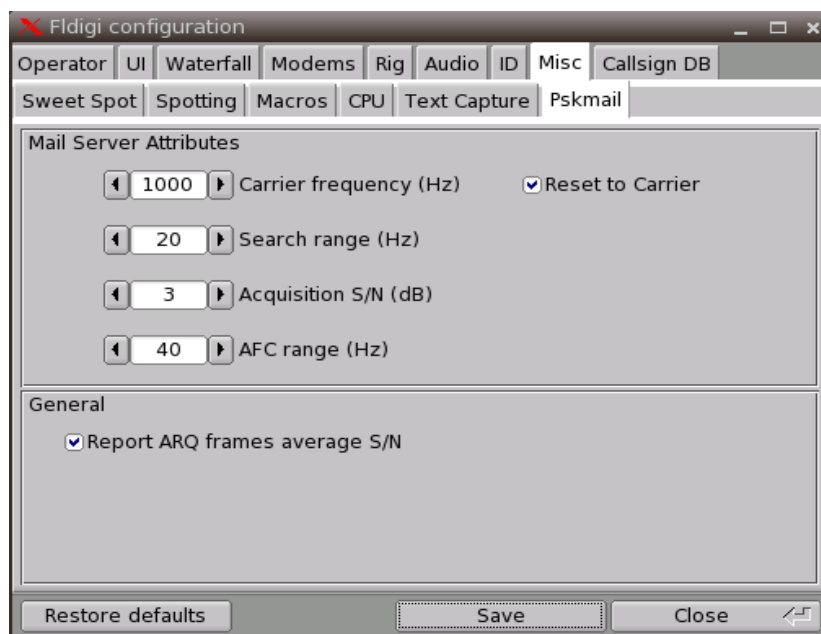
Carrier frequency is the audio frequency you wish to use with your transceiver. Many use 1000 Hz here as that makes frequency calculation easy. For instance if the server uses the assigned frequency 10147 and you enter 1000 Hz then the dial should say 10146 (upper sideband). It is important that the carrier frequency is in the center of your narrow filter. E.g, if the center of your dsp filter in an IC756 is on 1500 Hz, set your carrier frequency to 1500 Hz.

AFC range (Hz)

Its good to have the client and server stay on the same frequency, the modems will try to find each other by adjusting the frequency (AFC). That is why we limit how far away that AFC will be able to move the frequency. *AFC range* is the setting to take care of, a 50 Hz limit should be fine for wide modes. Narrow modes such as PSK 63 will require a more narrow setting, say 10 Hz.

Reset to Carrier and Report ARQ frames average S/N

Reset to Carrier kicks in when there is no signal present and forces the audio frequency back to the set carrier frequency, it is recommended to enable that. Report ARQ frames **MUST** be set.



These settings are only valid when Fldigi is connected to the pskmail client. If Fldigi drifts away from

the frequency and seems to ignore what you have entered here then perhaps the connection between the client and the modem was not successful. The initialization will only take place when the client or server connects to the modem (Fldigi). This means you should **always start Fldigi prior to starting the pskmail client or server**.

Please ensure:

- That Fldigi is started before the server or client is started.
- If Fldigi is restarted for any reason, the server or client must also be restarted.

RS ID

The next to take care of is RS ID settings, which is very simple. Just uncheck everything in the RS ID frame, illustration of that below:

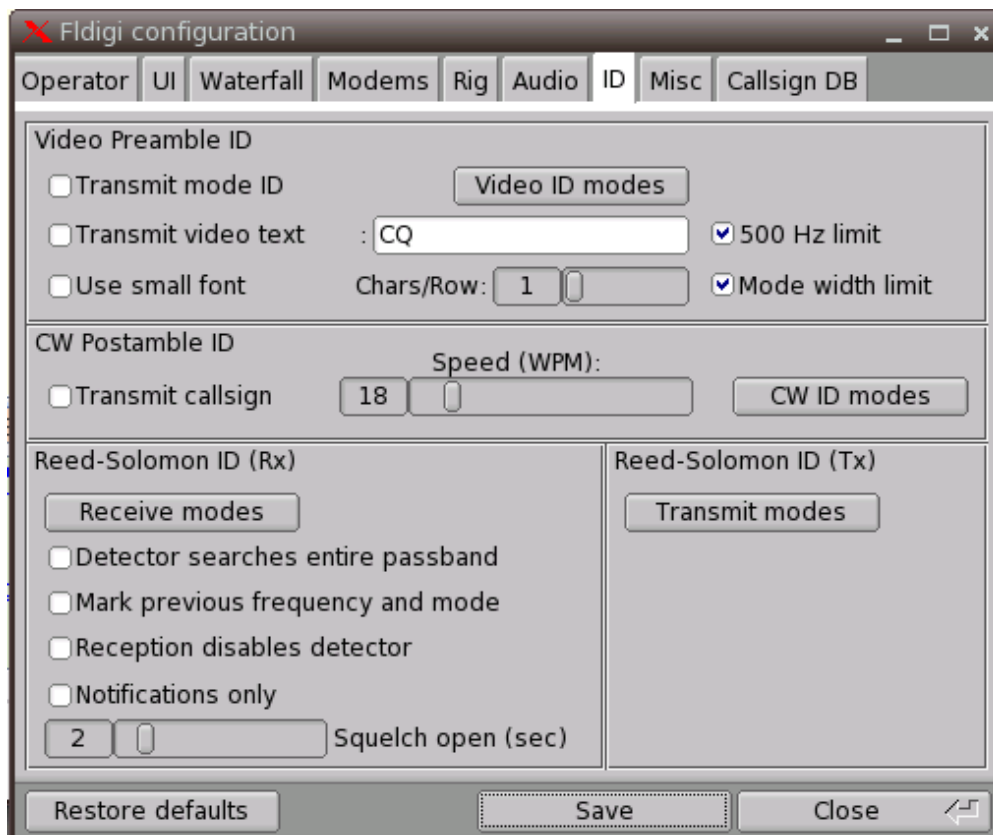
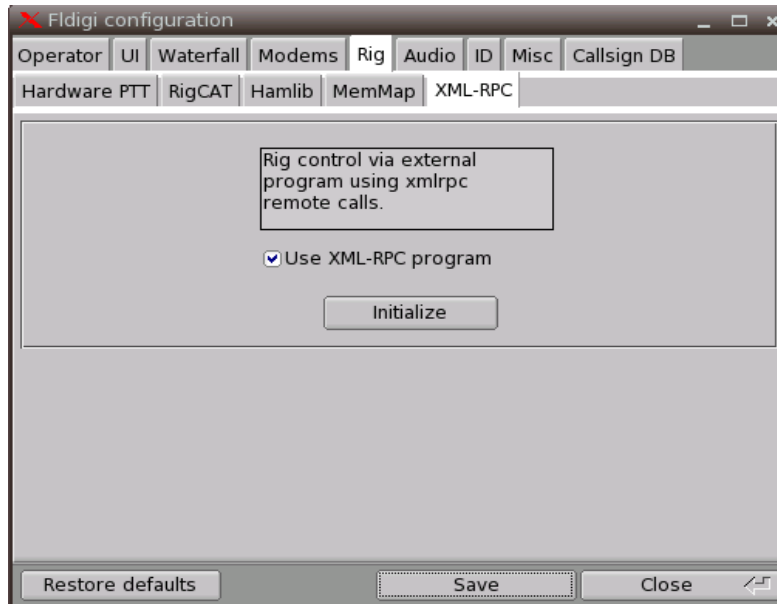


Illustration 1: RS ID settings

It is advantageous to disable RSID receive and transmit for modes which pskmail is not using, that makes sure fldigi does not switch to an invalid mode.

Also enable the XMLRPC port in fldigi, which is used for rig control and S/N reporting:



When Fldigi works you can try to send some pings on one of the server frequencies, if your signal is decoded by a server then you will get a reply.

In some configurations Fldigi will not send the last character of a frame (<EOT>). When this is the case you can add some random characters to the frame in the Modem Tab of the Edit>Preferences screen. Once you get answers from a server the preparations are done.

You could ask another station on frequency to take a look at the transmitted signal, make sure it is not too wide. The TX audio should just touch the ALC.

Installing the PSKmail server on UBUNTU

Connect to the internet...

Download the server from <http://hermes.esrac.ele.tue.nl/pskmail/> at the time of writing this, the last stable version is pskmail_server-1.0.34.tar.gz.

Copy this tar file to your user directory (*cp pskmail_server-1.0.34.tar.gz ~/*)

```
cd ~
```

```
tar zxvf pskmail_server-1.0.34.tar.gz
```

```
cd pskmail_server-1.0.34
```

```
./INSTALL
```

The INSTALL script will run the libinstall.sh and install.sh scripts, to make sure all required libraries are on board. You will be asked to give your root password... answer 'yes' to most other questions.

Getting Puppy Linux

1. Download the puppy distro: <http://hermes.esrac.ele.tue.nl/pskmail/liveCDs/lupu-511.iso>
2. Burn the iso to a CD
3. Boot from the CD
4. As soon as puppy runs, read the help files !!!!!!!!!!!!!!!
5. Configure internet access and browser
6. Reboot puppy
7. Make a save file (see help files)

Installing fldigi on Puppy Linux

For fldigi on puppy you need the following files from

http://hermes.esrac.ele.tue.nl/pskmail/puppy_pets

1. fl_libs.pet
2. fldigi-3.20.pet
3. fldigiconfig.pet

Install these pets in the right sequence... fldigi will also install hamlib for you.

After installing you can test if fldigi works with your transceiver.

Test PSK500 and THOR8.

If you want to use the scanning function, you should also test hamlib with rigctl.

Installing the server on puppy

For the server you need the following files from

http://hermes.esrac.ele.tue.nl/pskmail/puppy_pets

1. perl-5.10.pet
2. pskmail_server-1.0.34.pet
3. lynx_elinks.pet
4. openssh_server-5.1p1.pet
5. x11vnc_server-0.8.4B.pet
6. dclock-2.2.2.pet

Server configuration

After you have installed all the files it is time to change the server configuration.

The program files are in /usr/local/share/pskmail_server, and you don't need to touch them.

All configuration is done in the /root/.pskmail directory. You should configure

- pskmailrc.pl
- the freq.txt file in .pskmail/qrg

The pskmailrc.pl file

You must at least change the callsign, the smtp server data and the location data.

Here is an example of a typical pskmailrc.pl file, with comments:

```
#!/usr/bin/perl -w
#PSKmail server config data
$Iamserver = 1;
$ServerCall = "YOURCALL_IN_CAPITALS";
$Pingdelay = 0;
$Beaconminute = 3;
# The following array can contain 1's or 0's for minutes 0...4
# The on-air beacons will be sent if it is a '1', the following line means send beacons in minutes
# 0, 2 and 4.
@Beaconarray = qw (1 0 1 0 1); # depends on which minutes you want to beacon
$commandcall = "XXXXXX"; # could be your call
$relay = "yoursmtplibserver"; #put the address of your smtp server here, e.g. smtp.isp.org ...
$smtpauthlevel = 0; # This depends on the authorization level of your smtp server
# level 0 = no auth, level 1 = user/password, 2 = TLS
$smtpuser = ""; # This depends on your smtp server
# when you have to use a '@' it must be escaped like '\@'
# example: user\@ISP.com
$smtpspass = ""; # Password of your smtp server
$smtpsport = 25; # This depends on your smtp server, ask your ISP
$monitor = 1; # monitor packets in terminal
$ShowBlock = 1; # also monitor raw packets as they come from fldigi
$debug = 0;
$Max_retries = 10;
$Maxidle = 17;
$Maxwebpage = 30000; # max. length of webpage download, truncated after that
$Txdelay = 0; # When your Tx transmits on top of the received packets
$Framelength = 17; # fixed
$nosession = "none"; #fixed
$period = 60; # on-air beacon period, normally on top of the hour (mins 0...4)
$Emails_enabled = 1; # disable with 0
```

```

$Web_enabled = 1; #disable with 0
$dbfile = "$ENV{HOME}/.pskmail/rflinkusers/rflink_users.db"; # fixed
$output = ">$ENV{HOME}/.pskmail/gmfsk_autofile"; # deprecated
$TxInputfile = "$ENV{HOME}/.pskmail/TxInputfile"; # fixed
$Inputfile = "$ENV{HOME}/.pskmail/gMFSK.log"; # deprecated
$BigEarsserverport = 10148; # reserved
$Aprs_connect = 1; # Put 0 if you don't want to connect to aprs
$latitude = "0000.00N"; # your latitude in ddmm.mm
$longitude = "00000.00E"; # your longitude in dddmm.mm
$serverstatus = "PSKmail 1.0.34 www.pskmail.com";
$Aprs_beacon = "0000.00NP00000.00E&www.pskmail.com "; # your location, see above
@Aprs_port = qw (1314 1314 1314); # ports of tier2 aprs server
# In the next variable you can put other servers if you like...
@Aprs_address = qw (netherlands.aprs2.net germany.aprs2.net italy.aprs2.net);
@prefixes = qw(); #deprecated
$posit_time = 10; #deprecated
$scheduler = 0; # used for scheduling
$rigtype = 326; # hamlib rif number, depends on your trx
$rigrate = 9600; # hamlib port speed, depends on your trx
$rigdevice = "/dev/ttyS0"; # hamlib port, depends on your rigctl port
$scanner = "F"; # F = frequency scan, S = mode only scan.
$qrgfile = "$ENV{HOME}/.pskmail/qrg/freqs.txt"; # fixed
$freq_offset = 0;
@freq_corrections = qw (-1000 -1000 -1000 -1000 -1000); # for 1000 Hz offset
$traffic_qrg = "10147500"; # change for other QRG
$Defaultmode = "PSK500R";
$MARS = 1; # put 0 if you are in the US and not a MARS station. Will default to PSK250.
1;

```

The freq.txt file

The freq.txt file is in ~/.pskmail/qrg

A typical file looks like:

```

10147000,10147000,10147000,10147000,10147000,
PSK500R,PSK500R,PSK500R,PSK500R,PSK500R,
USB,USB,USB,USB,USB,
YOURCALL,YOURCALL,YOURCALL,YOURCALL,YOURCALL,

```

You can enter different values for the 5 scan minutes.

You can enter different calls for different bands, if your licence makes that necessary.

Running the server

1. Connect to the internet.
2. Start fldigi, and make sure it works ok with your transceiver.
3. Grab a terminal and start the server with 'pskmail_server'.

That should be all...

Running the server remotely

You can run the server remotely by first starting the x11vncserver.

Then start fldigi and the server from the remote computer via the TightVnc client.

The PI4TUE server has been running that way for 5 years without any problem.

Time keeping

It is a good idea to run **ntpd**, to keep the time exact. Puppy includes a Psync time synchronizer which you could run periodically from a cron job.

Implemented services

Several different services have been implemented on the pskmail server. They are normally invoked via commands from a pskmail client, but there are some services that operate in a different way.

Sending bulletins

You can make pskmail send a bulletin by copying a text file called 'bulletin' into /home/user/.pskmail/pskdownload/ directory through a cron job.

The pskmail server will check periodically if there is a bulletin to send. The file will be deleted after it has been sent. The default mode is used for sending the bulletin.

A bulletin starts with:

ZCZC

QTC de <SERVER>

Date xxxxx

and always ends with

NNNN

A client will look for these codewords and stores the bulletin into a file.

Sending long bulletins

Long bulletins work the same, but the file is called 'longbulletin'.

Long bulletins are sent in MFSK32 mode

Sending weather beacons

The server can send APRS weather data to APRS-IS instead of a status. This is invoked by copying a file named aprs_wx.txt to /home/user/.pskmail/

The server will use '_' as an aprs icon indicator to signal it has weather data.

Please see the aprs specification for the format of the text string...

The file will not be deleted after sending.

Server commands

Command	comments
~TEST	Sends file ~/.pskmail/testfile
~QSY	Server goes to \$traffic_qrg
~QSY xxxxxx	Server goes to frequency xxxxx
~STATUS?	Shows uptime and memory status
~CHANNELS	Shows which summoning channels are configured
~ROUTES	Shows which stations are linked to which server
~GETBEACONS	Shows how many beacons received for this client last 12 hours
~QTC xxx+	Request mail headers from nr. xxx
~MAIL	Get all mail for this client account
~LISTLOCAL	List local mail for client call
~DELETELOCAL x x x	Delete local mails x x x for client call
~READLOCAL x	Read local mail x for client call
~READZIP xxx	Read compressed mail xxx
~READ xxx	Read mail xxx plain
~KEEP x x x	Forward mails x x x to client isp account with [mailarchive] tag
~GETFILE xxx.x	Send file xxx.x from pskdownloads directory (plain)
~GETBIN xxx.x	Send file xxx.x from pskdownloads directory (compressed)
~FO5:	DTN Transaction offer (priority 5)
~FY:xxxxxxxxxxx	DTN Transaction accept tag xxxxxxxxxxx
~FA:xxxxxxxxxxx	Delete DTN Transaction with tag xxxxxxxxxxx
~FN:xxxxxxxxxxx	Cancel transaction with tag xxxxxxxxxxx
>FM:	Receive DTN transaction content (mail, file)
~GETIAC	Get IAC fleetcode file from web and send compressed
~LISTFILES	List files in pskdownload directory
~TGET <URL>	Get web page and send plain text
~TGETZIP <URL>	Get web page and send compressed
~GETPOS <call>	Get aprs position for call
~GETMSG <nr>	Get <nr> messages from findu.com
~GETNEAR	List aprs stations near this client

~GETTIDE xxxx	Get tide data for location xxxx
~GETTIDESTN	List location nrs for tide stations near this location
~GETSERVERS	Get list of pskmail servers from http://pskmail.wikispaces.com/PSKmailservers
~GETNEWS	Get news page from pskmail wiki
~GETWWW	Get latest propagation data from NOAA
~GETCAMP xx.xx xx.xx	Get RV camp sites near location xx.xx xx.xx
~GETRELAYS xx.xx xx.xx	Get VHF/UHF relays near position xx.xx xx.xx
~MSG CALL@SERVER m	Store message m in local mailbox for CALL
~TWEET message	Tweet message in pskmail twitter account
~GETUPDATE	Get last 20 updates from pskmail twitter account
~POSITION xx.xx xx.xx	Send position to findu
~QUIT	Close session
~DELETE x x x	Delete mails x x x at isp account
~TELNET host port	Start telnet session
~RECxxxxxxx	Receive account data
~CSEND	Receive compressed email
~SEND	Receive plain text email

Appendix A, Mode selection

This table tries to gather data about the datamodes that are usable with pskmail. Its important to note that this is an attempt and a work in progress, test results and data that can be used to enhance this table is most welcome!

Anyway, a few facts and conclusions can be drawn from this table:

- The fastest mode is PSK500, live tests have shown speeds of 1862 char/min with ARQ.
- The fastest mode for US hams on HF is PSK250
- The slowest mode here is THOR 8 at 28 wpm
- The most narrow mode is PSK63, 63 Hz wide

Mode	Baud rate	Speed wpm capital letters	Speed wpm small letters	Speed* bps	Bandw. (hz)	SNR at least	Choice (1-3) for				Description and use
							NVIS	Weak signals	High Speed	Flutter, fading	
PSK63	63	74	102	88	63	-2	3	3	3	3	Vy narrow bt also vy slow
PSK125	125	148	204	176	125	-1	3	3	3	3	Better
PSK250	250	296	408	352	250	-1	3	3	2	3	Good all round mode
PSK500	500	592	816	704	500	2	3	3	1	3	Very fast, best 4 good condx
PSKR125	125	74	102	88	125	-8	2	2	3	2	
PSKR250	250	148	204	176	250	-5	2	2	2	3	
PSKR500	500	296	408	352	500	-3	2	3	1	3	Best allround mode?
MFSK16	15,6	58	58	58	316	-8	1	1	3	1	
MFSK32	31,25	120	120	120	630	-6	1	1	2	1	
Dominoex 11	10.8	70	70	70	262	-7	3	2	3	3	fdigi implementation flawed?
Dominoex 16	15.6	100	100	100	355	-6	3	2	3	3	fdigi implementation flawed?
Dominoex 22	21.5	140	140	140	524	-5	3	2	2	3	fdigi implementation flawed?
THOR8	7.8	28	28	28	346	-6	1	1	3	2	
THOR11	10.7	40	40	40	262	-5	1	X	3	2	
THOR16	15.6	58	58	58	355		1	X	3	2	
THOR22	21.5	78	78	78	524		1	X	2	2	

* Speed is an attempt. Varicode and different ways to measure makes this difficult

** SNR is whatever I could find, most at 3 KHz bandwidth..

You may notice that certain popular modes are missing from this table. Two of the missing modes are Olivia and MT63. Olivia is not usable with pskmail as it doesn't support the full 8-bit character set, that means the <SOH> frame start and <EOT> frame end cannot be transferred.

Mode tables for adaptive asymmetric link control

As from jPSKmail version 0.5, separate adaptive mode control for up- and downlink modes is automatic.

To accomplish this a number of changes to server and client were necessary. John, VK2ETA has played an important role herein. Based on empirical data, two mode tables were put into the server, one for

use by US hams and one for the rest of the world. This was necessary because the FCC will not allow symbol rates > 300 Bd. And PSK500 and PSK500R use 500 Bd.

Both tables incorporate the most popular modes, covering a large range of speeds, sensitivities and robustness (qrm, doppler, multipath).

Table A:

PSK500 – PSK500R – PSK250R – MFSK32 – THOR22 – MFSK16 – THOR8

Table B (US and Intermar):

PSK250 – PSK250R – PSK125R – MFSK32 – THOR22 – MFSK16 – THOR8

The modes in these tables run from fast to slow, and become more robust and sensitive from right to left. The server knows the S/N ratio and the download success rate in the client, and chooses the modes which allow the best throughput for the present channel conditions. Initial tx and rx modes are set on the client prior to issuing a connect frame. This is (still) under operator control.

Before sending a frame the server decides if a mode change is possible (up) or necessary (down) based on the S/N ratios of both links and arq success history. In that case the server switches one position up or down in the table.

APRS Mode choice

For the APRS functions the best mode is normally the default channel mode. In Europe there are two nets. One on 10147 kHz listening in PSK500R mode, and one on 10148 kHz listening in PSK250 mode. US servers will also use PSK250, unless they operate on MARS frequencies.

In case a station does not come through in the default mode it is possible to use a lower speed mode with RSID on. TX RSID can be switched on manually in fldigi. On receipt of a QSL from a server the client's TXID will automatically be switched off again. This to prevent mode chaos on the frequency...

Appendix B:

PSKmail ARQ protocol specification (2nd draft)

Introduction

PSKmail is an ARQ data transmission system. It has been optimized for use on short wave (HF) radio frequencies. PSKmail has both a client/server architecture and a client/client architecture. The servers are gateways to diverse services on the internet. The clients enable mobile or portable radio stations to access the internet where there is no other coverage, or send messages and files to each other.

To use PSKmail, one uses a short wave radio and a PC with a sound card, and often a hardware interface between the radio and the PC. The communications protocol has been optimized for the specific conditions on short wave:

- Varying propagation conditions on short wave, with multi-path, doppler effects, and often weak signals near the noise floor.
- Often the mobile stations run low power and compromise antennas.
- As ham radio uses shared frequencies, interference levels are high.

To cater for this wide range of conditions, PSKmail does not use a single digital radio mode. Instead it uses a range of modes with **separate adaptive control on up- and download channels**. The maximum bandwidth of the HF system is 500 Hz, and speeds range from 44 CPS for ideal channels down to 2 CPS for the most problematic ones. For clean (single hop) channels PSK modes are used, for fuzzy channels MFSK and THOR modes are employed, including convolutional coding, Viterbi decoders, and interleavers to counteract static noise...

Radio interface

The radio is always used in USB mode, as several of the modes used are polarized. THOR modes, e.g., are derived from DominoEx. Often a hardware interface is needed between modem and radio, to provide galvanic isolation for audio signals and PTT signal. *The protocol timing also allows the use of VOX.*

Modem interface

PSKmail uses Fldigi (>= 3.20) as a modem, which is a multi-mode digital modem program using the computer sound card. PSKmail uses ASCII characters 0x01 (<SOH>, 0x04 <EOT>, 0x06 <ACK>, 0x12 <DC2> and printable ASCII characters 32 ... 127 to talk to the modem.

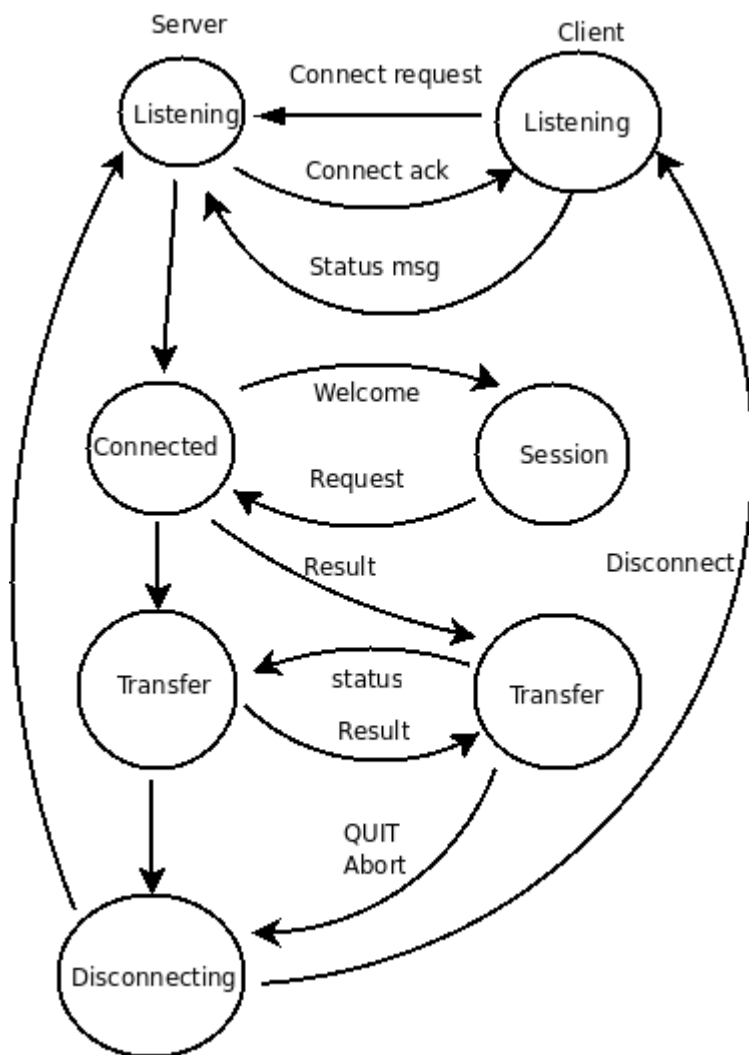
The system uses **RSID signaling** to notify the link layer protocol engine of mode

changes, and also makes signal/noise ratio data available to the application. PTT signaling is generated by Fldigi. The modem program is available for all operating systems.

Link layer protocol

The PSKmail system uses a master/slave protocol. The server is the master, the client is the slave. In case two clients connect each other for a TTY (chat) session, the caller is the slave. All control functions are in the master, which makes timing a lot easier than in the first PSKmail implementations, where client and server would compete until they were transmitting synchronously on top of each other.

Status diagram



Protocol details

All transactions between server and client are packed in **frames**. Every frame consists of maximum 10 **blocks**, viz. 0 ... 8 data blocks, 1 status block, and 1 optional identification block (at start of session and every 10 minutes thereafter).

Frame structure:

<SOH>[<ident block>]<data block>... <data block><status block><EOT>

All blocks have the same structure. They contain a header, a data partition, and a CRC16 number. The start of a block is marked with a <SOH> (ASCII 0x01) character. A status block ends with a <EOT> (ASCII 0x04) character. The data partition can be **0 ... 64 characters** long.

Block structure:

<SOH><header><data><CRC>

or

<SOH><header><data><CRC><EOT> (status block)

The **header** consists of 3 characters. The meaning of these characters is different for master (server) and slave (client), and also depending on block type. For the **client header**:

Block type	Character 1	Character 2	Character 3
Status	Rx s/n [0x20 - 0x7A]	Session [0x20 - 0x5F]	s
Data	0	Session [0x20 - 0x5F]	Blocknr. [0x20 - 0x5F]
Connect	Protocol version [0,1]	0	c
Summon	Protocol version [0,1]	0	n
Abort	0	Session [0x20 - 0x5F]	a
Ident	0	Session [0x20 - 0x5F]	i

For the **server header**:

Block type	Character 1	Character 2	Character 3
Status	Next Txmode [0 -8]	Session [0x20 - 0x5F]	s
Data	0	Session [0x20 - 0x5F]	Blocknr. [0x20 - 0x5F]
Connect ack	Protocol version [0,1]	Session [0x20 - 0x5F]	k
Disconnect	0	Session [0x20 - 0x5F]	d
Ident	0	Session [0x20 - 0x5F]	i

Some blocks have a **fixed data field**:

Block type	Data partition
Connect	PI4TUE:24 PA0R:1024 x (x = 0 – 8, initial server TX mode)
Connect ack	PA0R:1024 PI4TUE:24 x (x = block data length)
Summon	PI4TUE:24 PA0R:1024 10146000 x (x = 0 – 8, initial server TX mode)
Ident	PA0R de PI4TUE

ARQ control and repeats

The receive buffer is a circular buffer with room for 64 blocks. The **status block data field** carries the **status of the receive buffer**:

Char 1	Char 2	Char 3	Chars 4-12
Last block sent this frame	Buffer o.k. until/incl. block x	Last good block received	Missing blocks
[0x20 - 0x5F]	[0x20 - 0x5F]	[0x20 - 0x5F]	0 -8 x [0x20 - 0x5F]

Missing blocks are repeated by adding them to the next frame.

Adaptive mode control

The server controls both its own digital mode and that of the client.

The mode table covers a broad range of speeds and modes:

PSK500	PSK500R	PSK250R	MFSK32	THOR22	MFSK16	THOR8
---------------	----------------	----------------	---------------	---------------	---------------	--------------

For use in USA , where Baud rates over 300 Baud are not allowed on ham radio frequencies below 30 MHz, an alternative mode table is available:

PSK250	PSK250R	PSK125R	MFSK32	THOR22	MFSK16	THOR8
---------------	----------------	----------------	---------------	---------------	---------------	--------------

Both client and server measure S/N ratio of the frame received, and the client sends its SNR data to the server in every status block. The server signals mode changes to the client via RSID and through a flag in its status block header.

If a status block is not received, a poll frame is sent, which is just an empty frame without data blocks.

TX and RX modes are controlled in such a way that the number of ARQ repeats are limited, providing optimum throughput for the relative channel.

The server uses RSID on receive, so the client can connect in any of the available

modes. Default listening mode is PSK500R, but the server can scan more sensitive modes for DX if necessary.

Compression techniques

The PSKmail server uses **gzip** to compress web pages, emails and binary files. This zipped image is then **Base64** coded for transmission with the ASCII127 character set. Coding and decoding are automatic.

Unproto (non-ARQ) operation

There are 3 PSKmail functions which use unproto frames. These functions are:

- Ping
- Link to server
- APRS Beacons and Messaging

Like ARQ frames, unproto frames consist of a header, a data field and a CRC. When the client sends a **ping frame**, all servers answer with a **ping reply frame**.

Ping frame	<SOH>00uPA0R:7 ABCD<EOT>
Ping reply frame	<SOH>PI4TUE:71 BCDA<EOT>

Every PSKmail server has a **link table** containing client calls. A client can only be in the link table of 1 server. This function is used when the server sends an APRS message to a specific client. This way, the message is only sent by one server. The client sends a link request to a specific server, and on receipt of the request the server acknowledges the request:

Link request	<SOH>00uPA0R<>PI4TUE ABCD<EOT>
Link acknowledge	<SOH>00uPA0R<>PI4TUE ABCD<EOT>

The server and the client can send a **beacon on HF**, containing its APRS position and a status message:

Server beacon:	<SOH>00uIS0GRB-3:72 Pskmail_Server_0.9.34-10147.0KHz -DD6E<EOT>
Client beacon	<SOH>00uIK2YXT:26 !4507.95N/01044.65E-I1B80<EOT>

When the server hears a client beacon or an APRS message, it confirms reception with a short **QSL message**:

QSL message	<SOH>QSL PA0R de PI4TUE 555F<EOT>
-------------	-----------------------------------

An **APRS message** has the same format as the client beacon, an **APRS email** starts with an [address@something.xxx](#) .

APRS message	<SOH>00uDJ0LN:26 PA0R This is the APRS message 1B80<EOT>
APRS email	<SOH>00uDJ0LN:26 PA0R@ISP.ORG This is the email message 2A97<EOT>

CQ	<SOH>#00uPA0R:27 CQ CQ CQ PSKmail 02B2<EOT>
----	---

RSID signaling

RSID (Reed Solomon Identification) is used for mode change signaling. Thereby the following rules are applied:

	TX RSID	RX RSID
Client Idle (APRS)	OFF	OFF (manual switch ON)
Client receives acknowledge	OFF	OFF
Client connect request	ON	ON
Client data transfer	OFF	ON
Server Idle (APRS)	OFF	ON
Server sends acknowledge	ON	ON
Server data transfer	OFF	OFF
Server changes mode	ON	OFF

References

- MFSK modes <http://www.qsl.net/zl1bpu/MFSK/>
- PSK modes <http://aintel.bi.ehu.es/psk31.html>
- Fldigi descriptions <http://www.w1hkj.com/FldigiHelp-3.12/Modes/psk.htm>
<http://www.w1hkj.com/FldigiHelp-3.12/Modes/mfsk.htm>
<http://www.w1hkj.com/FldigiHelp-3.12/Modes/thor.htm>
- Fldigi <http://www.w1hkj.com/beginners.html>
- PSKmail: <http://www.pskmail.com>
- Mailing list: <http://www.freelists.org/archive/pskmail>